UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/681,860	10/07/2003	Andrew Crosland	ALTRP089/A987	7377

51501 7590 09/20/2006

BEYER WEAVER & THOMAS, LLP
ATTN: ALTERA
P.O. BOX 70250
OAKLAND, CA 94612-0250

EXAMINER	
LEE, CHRISTOPHER E	

ART UNIT	PAPER NUMBER
2112	

DATE MAILED: 09/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/681,860	Applicant(s) CROSLAND ET AL.	
	Examiner Christopher E. Lee	Art Unit 2112	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 09 August 2006.
2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-44 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☒ Claim(s) 4,14,15,19,20,24,25,29 and 30 is/are allowed.
6)☒ Claim(s) 1-3,5-13,16-18,21-23,26-28 and 31-44 is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a)☐ All b)☐ Some * c)☐ None of:
1.☐ Certified copies of the priority documents have been received.
2.☐ Certified copies of the priority documents have been received in Application No. _____.
3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Receipt Acknowledgement

1. Receipt is acknowledged of the request filed on 9th of August 2006 for a Request for Continued Examination (RCE) under 37 CFR 1.114 based on the Application No. 10/681,860, which the request is acceptable and an RCE has been established. Claims 1, 11, 21, and 31 have been amended; no claim has been canceled; and claims 33-44 have been newly added since the Final Office Action was mailed on 12th of April 2006. Currently, claims 1-44 are pending in this Application.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 1-3, 11-13, 16-18, 21-23, 26-28, 33, 35-37, 39, 40, 43, and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over LaBerge [US 6,108,778 A] in view of Lange [US 6,055,598 A].

Referring to claim 1, LaBerge discloses a method for handling resets at a bus bridge (i.e., blocking transactions at Controller 50 in Fig. 2 while computer system components are in reset; See col. 1, lines 10-13), the method comprising:

- providing a bus bridge (i.e., Controller 50 in Fig. 2) coupling a master component (i.e., requesting agent, e.g., Processor-N 10 in Fig. 2) supporting a master component bus protocol (i.e., P6 protocol) to a slave component (i.e., target agent, e.g., PCI-1 device 130 in Fig. 2) supporting a slave component bus protocol (i.e., PCI protocol; said Controller is bridging between said requesting agent and said target agent; See col. 3, lines 32-36);
- receiving a reset signal (i.e., receiving PCI Reset-1 at (F) in Fig. 2) at the bus bridge (i.e., said Controller), the reset signal associated with a reset of the slave component (i.e., said PCI Reset-1 being associated with a reset signal asserted by said PCI-N device; See col. 4, lines 3-14); and
- blocking to a transaction from the master component to the slave component with a bus protocol compliant signal (i.e., BNR# at (G) in Fig. 2) after the reset signal is received (See col. 1, lines 40-43 and col. 6, lines 49-63),
  - wherein responding with the bus protocol compliant signal (i.e., BNR# at (G) in Fig. 2) does not allow the master component (i.e., said requesting agent) to continue operating while the slave component (i.e., said target agent) is being reset (See col. 7, lines 3-17).

LaBerge does not teach said responding with said bus protocol compliant signal allows the master component to continue operating without violating bus protocols while the slave component is being reset, which means while the slave component is not ready to respond to said transaction from the master component to the slave component.

Lange discloses a method for allowing sequence-independent command responses across a computer bus bridge (i.e., PCI-to-PCI bridge device 100 in Fig.2; See col. 1, lines 7-11), wherein

- coupling a bus bridge master interface (i.e., secondary PCI master 112 and target bus interface 116 in Fig. 4) to a slave component (i.e., PCI target device, e.g., PCI option card 34 of Fig. 1);
- coupling a bus bridge slave interface (i.e., primary PCI slave 104 and initiating bus interface 102 in Fig. 4) to a master component (i.e., PCI command initiator, e.g., CPU 10 of Fig. 1), wherein a command retrying mechanism (See Fig. 5) performs
    - responding with a bus protocol compliant signal (i.e., RETRY signal for delaying the completion of the command cycle with PCI bus protocol) allows the master component (i.e., said PCI command initiator) to continue operating (i.e., to initiate other PCI commands) without violating bus protocols (i.e., delaying the completion of the command cycle without violating PCI bus protocol) while the slave component (i.e., said PCI target device) is not ready to respond to a transaction (i.e., PCI command) from the master component to the slave component (e.g., during a reset operation of said PCI target device; See col. 8, lines 19-28).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have included said command retrying mechanism, as disclosed by Lange, in said method for handling resets at a bus bridge, as disclosed by LaBerge, with coupling blocking signal (i.e., BNR#) to said command retrying mechanism so as to retry said transaction while said slave component is being reset (i.e., BNR# of said slave component is asserted as active state in Figs. 2 and 4), and thus said master component continues operating without violating bus protocols for the advantage of increasing overall data throughput (See Lange, col. 3, lines 11-15).

Referring to claim 2, LaBerge teaches

- the reset signal (i.e., PCI Reset-1 at Ⓕ in Fig. 2) is received at a bus bridge master interface (i.e., said PCI Reset-1 is received at PCI-1 Bus Logic 52 in Fig. 2).

Referring to claim 3, LaBerge teaches

- a bus bridge slave interface (i.e., Host Bus Logic 54 in Fig. 2) responds with the bus protocol compliant signal (i.e., said Host Bus Logic is blocked by BNR# signal in Fig. 3; actually, BNR# is blocking transactions on P6 Bus connected to said Host Bus Logic; See col. 7, lines 30-33).

Referring to claim 11, LaBerge teaches a method for independently resetting a side of a bus bridge (i.e., blocking transactions at Controller 50 in Fig. 2 while computer system components are in reset; See col. 1, lines 10-13, in fact, said resetting is independently performed by the respective computer system components at a side of said Controller), the method comprising:

- receiving a reset signal (i.e., receiving PCI Reset-1 at Ⓕ in Fig. 2) on a first side of the bus bridge (i.e., PCI-1 Clock Domain; See col. 3, line 54 through col. 4, line 2) such that a component associated with the first side (i.e., target agent, e.g., PCI-1 Device 130 in Fig. 2) is reset while a component associated with a second side of the bus bridge (i.e., Host Clock Domain; See col. 3, lines 47-53) continues operating without violating bus protocols,

  - the first side (i.e., PCI-1 Clock Domain) corresponding to a first clock domain (See Fig. 2 and col. 3, line 54 through col. 4, line 2) and
  - the second side (i.e., Host Clock Domain) corresponding to a second clock domain (See Fig. 2 and col. 3, lines 47-53).

LaBerge does not teach the component associated with the first side is reset, which means the component associated with the first side is not ready to respond to a transaction from the component associated with the second side to the component associated with the first side, while a component associated with a second side of the bus bridge continues operating without violating bus protocols.

Lange discloses a method for allowing sequence-independent command responses across a computer bus bridge (i.e., PCI-to-PCI bridge device 100 in Fig.2; See col. 1, lines 7-11), wherein

- coupling a second interface (i.e., secondary PCI master 112 and target bus interface 116 in Fig. 4) to a component associated with a first side (i.e., PCI target device, e.g., PCI option card 34 of Fig. 1);
- coupling a first interface (i.e., primary PCI slave 104 and initiating bus interface 102 in Fig. 4) to a component associated with a second side (i.e., PCI command initiator, e.g., CPU 10 of Fig. 1), wherein a command retrying mechanism (See Fig. 5) performs
  - the first interface (i.e., said primary PCI slave and initiating bus interface) configured to interact with the component associated with the second side (i.e., said PCI command initiator) such that the slave component (i.e., said PCI target device) is not ready to respond to a transaction (i.e., PCI command) from the component associated with the second side to the component associated with the first side (e.g., during a reset operation of said PCI target device) while the

master component (i.e., said PCI command initiator) continues operating (i.e., initiating other PCI commands) without violating bus protocols (i.e., delaying the completion of the command cycle without violating PCI bus protocol; See col. 8, lines 19-28).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have included said command retrying mechanism, as disclosed by Lange, in said method for handling resets at a bus bridge, as disclosed by LaBerge, with coupling blocking signal (i.e., BNR#) to said command retrying mechanism so as to retry said transaction while said slave component is being reset (i.e., BNR# of said slave component is asserted as active state in Figs. 2 and 4), and thus said master component continues operating without violating bus protocols for the advantage of increasing overall data throughput (See Lange, col. 3, lines 11-15).

Referring to claim 12, LaBerge teaches

- synchronizing the reset signal (i.e., PCI Reset-1 at Ⓕ in Fig. 2) with the second side of the bus bridge (i.e., Host Clock Domain; See Fig. 4 and col. 5, line 59 through col. 6, line 12) such that a component associated with the second side (i.e., requesting agent, e.g., Processor-N 10 in Fig. 2) can transition to an idle state (See Abstract, lines 4-5 and col. 7, lines 54-62; i.e., said Controller performs dynamically blocking transactions, viz., transition to an idle state, on the Host Bus).

Referring to claim 13, LaBerge teaches receiving the reset signal including

- receiving the reset signal (i.e., receiving PCI Reset-1 at Ⓕ in Fig. 2) at an interface (i.e., PCI-1 Bus Logic 52 in Fig. 2), the interface (i.e., said PCI-1 Bus Logic) coupling the first side component (i.e., target agent, e.g., PCI-1 Device 130 in Fig. 2) to the bus bridge (i.e., said PCI-1 Device is coupled to Controller 50 via said PCI-1 Bus Logic in Fig. 2).

Referring to claim 16, LaBerge teaches

- both the components are implemented in programmable logic (actually, Processor-N 10 as requesting agent is implemented in programmable logic, and PCI-1 device 130 as target agent is implemented in programmable logic in Fig. 2 since PCI transactions between said two agents are based on a configuration of programmable configuration spaces; See col. 3, lines 5-31).

Referring to claim 17, LaBerge teaches

- one of the components (i.e., Processor-N 10 in Fig. 2) is implemented in hard logic (i.e., being implemented in hardware components such as registers, combinational logic circuits, arithmetic logic, control circuit logic, etc., not a soft logic, e.g., software, firmware, and/or setting conceptual values; See col. 3, lines 5-11).

Referring to claim 18, LaBerge teaches synchronizing the reset signal (See col. 4, lines 38-50) including

- receiving a synchronization signal (i.e., output of Flip-Flop 82 in Fig. 3) by synchronization logic (i.e., Synchronization Logic 60 in Fig. 2), the synchronization signal is a result of the reset signal (i.e., PCI Reset-1 at (F) in Fig. 3; in fact, Flip-Flop 84 of said Synchronization Logic 60 receives said output of Flip-Flop 82 in Fig. 3);

- receiving a clock input (i.e., Host Clock at Ⓒ in Fig. 3) by the synchronization logic (i.e., said Synchronization Logic), the clock input associated with the second clock domain (i.e., Host Clock Domain; See Figs. 2 and 4, and col. 4, lines 38-41); and
- outputting an initialization idle signal (i.e., BNR# at Ⓖ in Fig. 3) when the received clock input indicates that the second clock domain is running (i.e., when said PCI Reset-1 is synchronized into said Host Clock; See Fig. 4 and col. 6, lines 55-62).

Referring to claim 21, LaBerge discloses a bus bridge (i.e., Controller 50 in Fig. 2) for handling resets (See Abstract), the bus bridge (i.e., said Controller) comprising:

- a bus bridge master interface (i.e., PCI-1 Bus Logic 52 in Fig. 2), the bus bridge master interface (i.e., said PCI-1 Bus Logic) coupled to a slave component (i.e., target agent, e.g., PCI-1 Device 130 in Fig. 2), the bus bridge master interface (i.e., said PCI-1 Bus Logic) configured to receive a reset signal (i.e., receiving PCI Reset-1 at Ⓕ in Fig. 2) associated with a reset of the slave component (i.e., said PCI Reset-1 being associated with a reset signal asserted by said PCI-N device; See col. 4, lines 3-14) on a first side of the bus bridge (i.e., PCI-1 Clock Domain; See col. 3, line 54 through col. 4, line 2);
- a bus bridge slave interface (i.e., Host Bus Logic 54 in Fig. 2), the bus bridge slave interface (i.e., said Host Bus Logic) coupled to a master component (i.e., requesting agent, e.g., Processor-N 10 in Fig. 2), wherein a device for blocking bus transaction during reset (See Abstract) performs
    - the bus bridge slave interface (i.e., said Host Bus Logic) configured to interact with the master component (i.e., said Processor-N) on a second side of the bus bridge (i.e., Host Clock Domain; See col. 3, lines 47-53) such that the slave component (i.e., said PCI-1 device) is reset while the master component (i.e.,

said Processor-N) is prevented from issuing a transaction (See col. 1, lines 40-43 and col. 6, lines 49-63).

LaBerge does not teach that the slave component is reset, which means the slave component is not ready to respond to a transaction from the master component to the slave component, while the master component continues operating without violating bus protocols.

Lange discloses a computer bus-to-bus bridging architecture (i.e., PCI-to-PCI bridge device 100 in Fig.2; See col. 1, lines 7-11), wherein

- a bus bridge master interface (i.e., secondary PCI master 112 and target bus interface 116 in Fig. 4), the bus bridge master interface (i.e., said secondary PCI master and target bus interface) coupled to a slave component (i.e., PCI target device, e.g., PCI option card 34 of Fig. 1);
- a bus bridge slave interface (i.e., primary PCI slave 104 and initiating bus interface 102 in Fig. 4), the bus bridge slave interface (i.e., said primary PCI slave and initiating bus interface) coupled to a master component (i.e., PCI command initiator, e.g., CPU 10 of Fig. 1), wherein a command retrying mechanism (See Fig. 5) performs
    - the bus bridge slave interface (i.e., said primary PCI slave and initiating bus interface) configured to interact with the master component (i.e., said PCI command initiator) such that the slave component (i.e., said PCI target device) is not ready to respond to a transaction (i.e., PCI command) from the master component to the slave component (e.g., during a reset operation of said PCI target device) while the master component (i.e., said PCI command initiator) continues operating (i.e., initiating other PCI commands) without violating bus protocols (i.e., delaying the completion of the command cycle without violating PCI bus protocol; See col. 8, lines 19-28).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have included said command retrying mechanism, as disclosed by Lange, in said bus bridge (i.e., Controller), as disclosed by LaBerge, with coupling blocking signal (i.e., BNR#) to said command retrying mechanism so as to retry said transaction while said slave component is being reset (i.e., BNR# of said slave component is asserted as active state in Figs. 2 and 4), and thus said master component continues operating without violating bus protocols for the advantage of increasing overall data throughput (See Lange, col. 3, lines 11-15).

Referring to claim 22, LaBerge teaches

- the first side (i.e., PCI-1 Clock Domain) corresponds to a first clock domain (See Fig. 2 and col. 3, line 54 through col. 4, line 2) and the second side (i.e., Host Clock Domain) corresponds to a second clock domain (See Fig. 2 and col. 3, lines 47-53),
- the first and second clock domains being different from each other (i.e., said PCI-1 Clock Domain's clock frequency is different from said Host Clock Domain's clock frequency; See col. 3, lines 39-46).

Referring to claim 23, LaBerge teaches

- a synchronization logic (i.e., Synchronization Logic 60 in Fig. 2) for synchronizing the reset signal (i.e., PCI Reset-1 at Ⓕ in Fig. 2) with the second clock domain (i.e., Host Clock Domain; See Fig. 4 and col. 5, line 59 through col. 6, line 12) such that the master component (i.e., requesting agent, e.g., Processor-N 10 in Fig. 2) and the bus bridge slave interface (i.e., Host Bus Logic 54 in Fig. 2) transition to an idle state (See Abstract, lines 4-5 and col. 7, lines 54-62; i.e., said Controller performs dynamically blocking transactions, viz., transition to an idle state, on the Host Bus).

Referring to claim 26, LaBerge teaches

- both the components are implemented in programmable logic (actually, Processor-N 10 as requesting agent is implemented in programmable logic, and PCI-1 device 130 as target agent is implemented in programmable logic in Fig. 2 since PCI transactions between said two agents are based on a configuration of programmable configuration spaces; See col. 3, lines 5-31).

Referring to claim 27, LaBerge teaches

- one of the components (i.e., Processor-N 10 in Fig. 2) is implemented in hard logic (i.e., being implemented in hardware components such as registers, combinational logic circuits, arithmetic logic, control circuit logic, etc., not a soft logic, e.g., software, firmware, and/or setting conceptual values; See col. 3, lines 5-11).

Referring to claim 28, LaBerge teaches the synchronization logic (i.e., Synchronization Logic 60 in Fig. 2) is configured

- for receiving a synchronization signal (i.e., output of Flip-Flop 82 in Fig. 3) that is a result of the reset signal (i.e., PCI Reset-1 at (F) in Fig. 3; in fact, Flip-Flop 84 of said Synchronization Logic 60 receives said output of Flip-Flop 82 in Fig. 3);
- for receiving a clock input (i.e., Host Clock at (C) in Fig. 3) that is associated with the second clock domain (i.e., Host Clock Domain; See Figs. 2 and 4, and col. 4, lines 38-41); and
- for outputting an initialization idle signal (i.e., BNR# at (G) in Fig. 3) when the received clock input indicates that the second clock domain is running (i.e., when said PCI Reset-1 is synchronized into said Host Clock; See Fig. 4 and col. 6, lines 55-62).

Referring to claim 33, LaBerge teaches

- either the master component (i.e., requesting agent Processor-N 10 in Fig. 2) or the slave component (i.e., target agent PCI-1 device 130 in Fig. 2) is implemented in programmable logic (actually, both of said Processor-N as said requesting agent and said PCI-1 device as said target agent are implemented in programmable logic, i.e., PCI transactions are based on a configuration of programmable configuration spaces) on a programmable device (i.e., said Processor and said PCI device are programmable devices; See col. 3, lines 5-31).

Referring to claims 35 and 36, Lange teaches

- responding with a bus protocol compliant signal (i.e., RETRY signal for delaying the completion of the command cycle with PCI bus protocol) allows the master component (i.e., PCI command initiator CPU 10 of Fig. 1) to continue operating (i.e., to initiate other PCI commands) without stalling nor locking up while the slave component is being reset (i.e., delaying the completion of the command cycle in order to avoid stalling or locking up during an incomplete operation of PCI target device, e.g., reset operation; See col. 8, lines 15-28).

Referring to claim 37, LaBerge teaches

- either the component associated with the first side (i.e., requesting agent Processor-N 10 in Fig. 2) or the component associated with the second side (i.e., target agent PCI-1 device 130 in Fig. 2) is implemented in programmable logic (actually, both of said Processor-N as said requesting agent and said PCI-1 device as said target agent are implemented in programmable logic, i.e., PCI transactions are based on a configuration of programmable configuration spaces) on a programmable device (i.e., said Processor and said PCI device are programmable devices; See col. 3, lines 5-31).

Referring to claims 39 and 40, Lange teaches

- responding with a bus protocol compliant signal (i.e., RETRY signal for delaying the completion of the command cycle with PCI bus protocol) allows the component associated with a second side (i.e., PCI command initiator CPU 10 of Fig. 1) to continue operating (i.e., to initiate other PCI commands) without stalling nor locking up while the first side is being reset (i.e., delaying the completion of the command cycle in order to avoid stalling or locking up during an incomplete operation of PCI target agent, e.g., reset operation; See col. 8, lines 15-28).

Referring to claims 43 and 44, Lange teaches

- the bus bridge slave interface (i.e., primary PCI slave 104 and initiating bus interface 102 in Fig. 4) is configured to interact with the master component on the second side of the bus bridge (i.e., PCI command initiator CPU 10 of Fig. 1) such that the slave component (i.e., PCI target device PCI option card 34 of Fig. 1) is reset while the master component continues operating without stalling nor locking up (i.e., delaying the completion of the command cycle in order to avoid stalling or locking up during an incomplete operation of PCI target device, e.g., reset operation; See col. 8, lines 15-28).

5. Claims 5-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over LaBerge [US 6,108,778 A] in view of Lange [US 6,055,598 A] as applied to claims 1-3, 11-13, 16-18, 21-

23, 26-28, 33, 35-37, 39, 40, 43, and 44 above, and further in view of Stewart et al. [US 6,687,773 B1; hereinafter Stewart].

Referring to claim 5, LaBerge, as modified by Lange, discloses all the limitations of the claim 5, including the master component bus protocol (i.e., P6 protocol) operating in a master component clock domain (i.e., Host Clock Domain; See Fig. 2 and col. 3, lines 47-53), except that does not teach the master component bus protocol is the AHB protocol.

Stewart discloses a bridge for coupling digital signal processor to on-chip bus as master (See Abstract), wherein

- a master component bus protocol (i.e., protocol for AHB Master on-chip Bus 30 in Fig. 1) is the AHB protocol (i.e., AMBA AHB protocol; See col. 4, lines 3-6 and 15-16) operating in a master component clock domain (i.e., AMBA AHB clock boundary; See col. 4, lines 40-43).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have substituted buses for master component and slave component, as disclosed by LaBerge, as modified by Lange, by said AHB buses with said AMBA AHB protocol, as disclosed by Stewart, for the advantage of providing a high performance proprietary core integration (See Stewart, col. 2, lines 7-32).

Referring to claim 6, LaBerge, as modified by Lange and Stewart, teaches

- the slave component bus protocol (i.e., protocol for AHB Slave on-chip Bus 32 in Fig. 1; Stewart) is the AHB protocol (i.e., AMBA AHB protocol; See Stewart, col. 4, lines 3-6 and 15-16) operating in a slave component clock domain different from the master component clock domain (i.e., PCI-1 Clock Domain's clock frequency is different from Host Clock Domain's clock frequency; See LaBerge, col. 3, lines 39-46).

Referring to claim 7, LaBerge, as modified by Lange and Stewart, discloses the slave component bus protocol (i.e., protocol for PCI-1 Bus 110 coupled to PCI-1 Bus Logic 52 in Fig. 2) is the PCI protocol (See col. 3, lines 32-38) instead of the PI protocol (See col. 3, lines 24-28; in fact, PCI bus protocol is not intended to be anyway so limited).

However, the claim merely recites the slave component bus protocol being the PI protocol without any patentable advantage in the specification.

Therefore, the limitations of the PI protocol in the claim is not patentably significant since it at most relates to the kind of bus protocols consideration which is not ordinarily a matter of invention. *In re Yount, 36 C.C.P.A. (Patents) 775, 171 F2.2d 317, 80 USPQ 141.*

Referring to claim 8, LaBerge teaches

- synchronizing the reset signal (i.e., PCI Reset-1 at Ⓕ in Fig. 2) to the master component clock domain (i.e., Host Clock Domain; See Fig. 4 and col. 5, line 59 through col. 6, line 12).

Referring to claim 9, LaBerge teaches

- setting the master component (i.e., requesting agent, e.g., Processor-N 10 in Fig. 2) to an idle state (See Abstract, lines 4-5 and col. 7, lines 54-62; i.e., dynamically blocking transactions, which means setting to an idle state, viz., no transactions).

Referring to claim 10, LaBerge teaches

- setting a bus bridge slave interface (i.e., Host Bus Logic 54 in Fig. 2) in an idle state (i.e., dynamically blocking transactions, which means setting to an idle state, viz., no transactions) while the slave component (i.e., PCI-1 Device 130 in Fig. 2) is being reset (i.e., said Host Bus Logic is blocked by BNR# signal in Fig. 3; actually, BNR# is blocking transactions on P6 Bus connected to said Host Bus Logic, i.e., no transactions - idle state, while said PCI-1 Device is reset by PCI-1 Reset in Fig. 2; See col. 7, lines 30-33).

6. Claims 31 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over LaBerge [US 6,108,778 A] in view of Lange [US 6,055,598 A] and what was well known in the art, as exemplified by Luo et al. [US 6,265,885 B1; hereinafter Luo].

Referring to claim 31, all of the claim limitations have already been discussed/addressed with respect to claim 1, with the exception of a computer readable recording medium storing program instructions for handling said method of claim 1 (e.g., memory having a computer software program for handling said method in claim 1).

The Examiner takes Official Notice that said method in the claim 1 being implemented in a program providing program instructions (i.e., a computer software program), and being stored in a computer readable recording medium (i.e., memory RAM, ROM, etc), is well known to one of ordinary skill in the art, as evidenced by Luo (See Claim 9, lines 2-4).
Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have implemented said method of the claim 1, as disclosed by LaBerge, as modified by Lange, in said program providing program instructions (i.e., a computer software program), and being stored in a computer readable recording medium (i.e., memory RAM, ROM, etc), as disclosed by Luo, since it would have provided a better flexibility of implementing said method than a hardware implementation, such as an easy modification, etc.

Referring to claim 32, LaBerge teaches

- the reset signal (i.e., PCI Reset-1 at (F) in Fig. 2) is received at a bus bridge master interface (i.e., said PCI Reset-1 is received at PCI-1 Bus Logic 52 in Fig. 2).

7. Claims 34, 38, 41, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over LaBerge [US 6,108,778 A] in view of Lange [US 6,055,598 A] as applied to claims 1-3, 11-13, 16-18, 21-23, 26-28, 33, 35-37, 39, 40, 43, and 44 above, and further in view of Eaton et al. [US 6,552,410 B1; hereinafter Eaton].

Referring to claims 34 and 38, LaBerge, as modified by Lange, discloses all the limitations of the claims 34 and 38, respectively, except that does not teach the programmable device is selected from the group consisting of a PLD, a PLA, a PAL, a FPGA, and a CPLD.

Eaton discloses a programmable antifuse interfacing a programmable logic and a dedicated device (See Abstract), wherein

- a programmable device (i.e., integrated circuit 410 with PCI Core 414 and user circuitry 412 in Fig. 16) is a FPGA (See col. 12, lines 30-34).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have used said programmable device, as disclosed by Eaton, for implementing said master and slave components, as disclosed by LaBerge, as modified by Lange, for the advantage of providing a user with the guaranteed functionality, ease-of-use, and high performance found in a dedicated device, as well as the configurability and flexibility found in programmable logic (See Eaton, col. 3, lines 54-59).

Referring to claim 41, LaBerge, as modified by Lange, discloses all the limitations of the claim 41, except that does not teach the bus bridge is implemented on a programmable device.

Eaton discloses a programmable antifuse interfacing a programmable logic and a dedicated device (See Abstract), wherein

- a bus bridge (i.e., integrated circuit 410 with PCI Core 414 and user circuitry 412 in Fig. 16) is implemented on a programmable device (See col. 12, lines 30-34).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have used said programmable device, as disclosed by Eaton, for implementing said master and slave interfaces, as disclosed by LaBerge, as modified by Lange, for the advantage of providing a user with the guaranteed functionality, ease-of-use, and high performance found in a dedicated device, as well as the configurability and flexibility found in programmable logic (See Eaton, col. 3, lines 54-59).

Referring to claim 42, Eaton teaches

- the programmable device (i.e., integrated circuit 410 with PCI Core 414 and user circuitry 412 in Fig. 16) is a FPGA (See col. 12, lines 33-34).

Allowable Subject Matter

8. Claims 4, 14, 15, 19, 20, 24, 25, 29, and 30 are allowed.

9. The following is a statement of reasons for the indication of allowable subject matter:

With respect to claim 4, the claim limitations are deemed allowable over the prior art of record as the prior art fails to teach or suggest that the bus protocol compliant signal is an AHB bus error signal.

With respect to claims 14 and 24, the claim limitations of the respective claims 14 and 24 are deemed allowable over the prior art of record as the prior art fails to teach or suggest that an alternating state handler being associated with the first side, wherein both the bus bridge master interface and the alternating state handler are configured for synchronously receiving the reset signal on the first side of the bus bridge.

The claim 15 is a dependent claim of the claim 14.

The claim 25 is a dependent claim of the claim 24.

With respect to claims 19 and 29, the claim limitations of the respective claims 19 and 29 are deemed allowable over the prior art of record as the prior art fails to teach or suggest that the bus bridge slave interface is configured for outputting a component idle signal to the master component in response to receiving the initialization idle signal, the component idle signal being used to trigger the master component to transition to an idle state.

The claim 20 is a dependent claim of the claim 19.

The claim 30 is a dependent claim of the claim 29.

Response to Arguments

10. Applicants' arguments filed on 9th of August 2006 have been fully considered but they are not persuasive.

In response to the Applicants' argument with respect to "... In the Office Action, the Examiner admitted that 'LaBerge does not teach said responding with said bus protocol compliant signal allows the master component to continue operating while the slave component is being reset.' (See Office Action page 3, line 20-21) The Examiner also stated that Lange discloses a command retrying mechanism that performs the step of responding with a bus protocol compliant signal (i.e., RETRY signal). It is respectfully submitted, however, that the RETRY signal is not a bus protocol compliant signal in the manner claimed. In particular, Lange does not teach or suggest that the RETRY signal allows the master component to continue operating without violating bus protocols while the slave component is being reset. ..." in the Response page 10, line 13 through page 11, line 2, the Examiner respectfully disagrees.

In contrary to the Applicants' allegation, LaBerge teaches that responding with the bus protocol compliant signal (i.e., BNR# at (G) in Fig. 2) does not allow the master component (i.e., requesting agent) to continue operating while the slave component (i.e., target agent) is being reset (See LaBerge, col. 7, lines 3-17). However, Lange teaches that responding with a bus protocol compliant signal (i.e., RETRY signal for delaying the completion of the command cycle with PCI bus protocol) allows the master component (i.e., said PCI command initiator) to continue operating (i.e., to initiate other PCI commands) without violating bus protocols (i.e., delaying the completion of the command cycle without violating PCI bus protocol) while the slave component (i.e., said PCI target device) is not ready to respond to a transaction (i.e., PCI command) from the master component to the slave component (e.g., during a reset operation of said PCI target device; See Lange, col. 8, lines 19-28). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have included a command retrying mechanism, as disclosed by Lange, in the method for handling resets at a bus bridge, as disclosed by LaBerge, with coupling blocking signal (i.e., BNR#) to said command retrying mechanism so as to retry said transaction while said slave component is being reset (i.e., BNR# of said slave component is asserted as active state in Figs. 2 and 4), and thus said master component continues operating without violating bus protocols for the advantage of increasing overall data throughput (See Lange, col. 3, lines 11-15). In other words, the combination of LaBerge and Lange with rationale for the proper combining suggests the obviousness of the claimed invention.

And, the Applicants' argument fails to comply with 37 CFR 1.111(b) because it amounts to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. In this case, the Applicants merely argue that the RETRY signal of Lange is not a bus protocol compliant signal in the manner claimed without specifically pointing out how the language of the claims "a bus protocol compliant signal" patentably distinguishes the claims from Lange reference. Furthermore, it is noted that the features upon which applicants rely (i.e., the RETRY signal allows the master component to continue operating without violating bus protocols while the slave component is being reset) are not recited in the rejected claims. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Actually, the exemplary claim 1 recites the limitation "***responding*** with the bus protocol compliant signal (i.e., RETRY signal) **allows** the master component to continue operating without violating bus protocols while the slave component is being reset" in lines 9-11. The Applicants have never recited the asserted recitation "***the bus protocol compliant signal*** *(i.e., RETRY signal)* **allows** the master component to continue operating without violating bus protocols while the slave component is being reset."

Thus, the Applicants' argument on this point is not persuasive.

In response to the Applicants' argument with respect to "Nonetheless, there is no motivation to combine the teachings of Lange with the teachings of LaBerge. This is because LaBerge teach away from allowing the master component to continue operating, much less having it operate without violating bus protocols while the slave component is being reset. ... In other words, LaBerge teaches preventing or blocking the master component from operating during reset. Furthermore, the command retrying mechanism and the RETRY signal are unlikely to work with the device (as disclosed in LaBerge) for blocking bus transactions during reset. This is because there is no indication in Lange that the command retrying mechanism and the

RETRY signal are adapted for a resetting. ..." in the Response page 11, lines 3-15, the Examiner believes that the Applicants misinterpret the claim rejection.

In fact, LaBerge discloses blocking to a transaction from the master component to the slave component with a bus protocol compliant signal (i.e., BNR# at (G) in Fig. 2) after the reset signal is received (See LaBerge, col. 1, lines 40-43 and col. 6, lines 49-63), wherein responding with the bus protocol compliant signal (i.e., BNR# at (G) in Fig. 2) does not allow the master component (i.e., requesting agent) to continue operating while the slave component (i.e., target agent) is being reset (See LaBerge, col. 7, lines 3-17) for dynamically delaying transactions on the master component until the slave component is out of reset (See LaBerge, Abstract). However, Lange discloses a method for allowing sequence-independent command responses across a computer bus bridge (i.e., PCI-to-PCI bridge device 100 in Fig.2; See Lange, col. 1, lines 7-11), wherein a command retrying mechanism (See Lange, Fig. 5) performs responding with a bus protocol compliant signal (i.e., RETRY signal for delaying the completion of the command cycle with PCI bus protocol) allows the master component (i.e., said PCI command initiator) to continue operating (i.e., to initiate other PCI commands) without violating bus protocols (i.e., delaying the completion of the command cycle without violating PCI bus protocol) while the slave component (i.e., said PCI target device) is not ready to respond to a transaction (i.e., PCI command) from the master component to the slave component (e.g., during a reset operation of said PCI target device; See Lange, col. 8, lines 19-28). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have included said command retrying mechanism, as disclosed by Lange, in said method for handling resets at a bus bridge, as disclosed by LaBerge, with coupling blocking signal (i.e., BNR#) to said command retrying mechanism so as to retry said transaction while said slave component is being reset (i.e., BNR# of said slave component is asserted as active state in Figs. 2 and 4), and thus said master component continues operating without violating bus protocols for the advantage of increasing overall data throughput (See Lange, col. 3, lines 11-15).

Furthermore, the Examiner notes that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Thus, the Applicants' argument on this point is not persuasive.

In response to the Applicants' argument with respect to "... For example, claims 16 and 26 recite the limitation of 'wherein both the components are implemented in programmable logic ...' This is in contrast to the Examiner's reference in the Office Action on page 3, lines 8-9, to 'PCI transactions being based on a configuration of programmable configuration space.'" in the Response page 11, lines 16-28, the Examiner respectfully disagrees.

As the Office Action on page 3, lines 8-9, PCI transactions are processed based on a configuration of programmable configuration space, which clearly supports that both the components, i.e., Processor-N and PCI device, are implemented in programmable logic, i.e., said Processor-N is programmable, and further, said PCI-device is configured under the PCI programmable configuration for the PCI transactions between said Processor-N and said PCI-device.

Thus, the Applicants' argument on this point is not persuasive.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Christopher E. Lee whose telephone number is 571-272-3637. The examiner can normally be reached on 9:30am - 5:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rehana Perveen can be reached on 571-272-3676. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Christopher E. Lee
Primary Patent Examiner
Art Unit 2112

CEL/

Christopher E. Lee